How do I respond to the WM_MENUCHAR message?

devblogs.microsoft.com/oldnewthing/20171208-00

December 8, 2017

Raymond Chen

Today, for no particular reason, I'll demonstrate how to handle the `WM_MENUCHAR` message.

The system sends you the message when a menu is active and the user presses a key that does not correspond to any mnemonic or accelerator key. The typical case for needing to handle this message is when you have a bitmap menu item and want to provide a mnemonic for it.

Let's take our scratch program and make these changes:

```
BOOL
OnCreate(HWND hwnd, LPCREATESTRUCT lpcs)
{
 HICON hico = LoadIcon(nullptr, IDI_INFORMATION);
 ICONINFO ii;
 GetIconInfo(hico, &ii);
 DeleteBitmap(ii.hbmMask);
 ModifyMenu(GetMenu(hwnd), 103, MF_BYCOMMAND | MF_BITMAP,
            103, (LPCTSTR)ii.hbmColor);
 return TRUE;
}

BOOL
InitApp(void)
{
 ...
 wc.lpszMenuName = MAKEINTRESOURCE(1);
 ...
}

// resource file
1 MENU
BEGIN
  POPUP "&File"
  BEGIN
    MENUITEM "&Something", 101
    MENUITEM "&Blah", 102
    MENUITEM "", 103
    MENUITEM "E&xit", 104
  END
END
```

We added a simple menu to our scratch program consisting of four options. Three are textual menu options: “Something”, “Blah”, and “Exit”. One is a bitmap, which we initialize in the `OnCreate` method to be the informational icon.

Note that we leak the bitmap. This is just a demonstration.

Note also that we don’t work particularly hard at scaling the bitmap to match the user’s screen DPI. This is just a demonstration.

Note further that we don’t do anything with the mask, so the transparent parts of the icon will show up as black. This is just a demonstration.

The program doesn’t do anything in response to the menu items. That’s not the point of the demonstration.

When you open the File menu, observe that you can use the keyboard shortcuts S, B, and X to invoke the corresponding menu items. The system recognizes the `&` prefix as indicating which character you want to treat as the mnemonic for that menu item.

But there is no keyboard shortcut for the information icon. It's just a bitmap. The system has no insight into the contents of the bitmap and come up with some sort of suitable mnemonic.

Let's fix that. Let's say that the mnemonic for the information icon should be I.

```
int GetMenuItemPos(HMENU hmenu, UINT id)
{
 int pos;
 for (pos = GetMenuItemCount(hmenu) - 1; pos >= 0; pos--) {
  if (GetMenuItemID(hmenu, pos) == id) break;
 }
 return pos;
}

DWORD OnMenuChar(HWND hwnd, UINT ch, UINT flags, HMENU hmenu)
{
 if ((flags & MF_POPUP) &&
     (hmenu == GetSubMenu(GetMenu(hwnd), 0))) {
  switch (ch) {
  case 'I':
   return MAKELONG(GetMenuItemPos(hmenu, 104), MNC_SELECT);
  }
 }
 return MAKELONG(0, MNC_IGNORE);
}
```

```
    HANDLE_MSG(hwnd, WM_MENUCHAR, OnMenuChar);
```

When we get the `WM_MENUCHAR` message, we verify that the current menu is the File menu by comparing the menu handles. (In a real program, we probably would cache the popup menu handle so we didn't have to digging for it each time.) If so, then we say that we want to select the information icon by returning a value that is a combination of

- The value `MNC_SELECT`, indicating that we want the item to be selected, and
- The index of the menu item to select.

To get the index of the menu item, we use a helper function `GetMenuItemPos` which looks up the position of a menu item given its ID.

With this change, you can how press I to select the information icon. It doesn't execute the item, though; it merely selects it. You could do this if, say, there were two information icons and you wanted to let the user cycle between them, the same way the selection cycles among multiple textual menu items with the same mnemonic.

If you want pressing I to invoke the menu item, then just make this change:

```
   return MAKELONG(GetMenuItemPos(hmenu, 104), MNC_EXECUTE);
```

Note that the `WM_MENUCHAR` message is sent only if the user presses a key that matches no textual menu item. If we change the last menu item to

```
    MENUITEM "Ex&it", 104
```

so that its accelerator is also `I`, then pressing the `I` goes straight to that menu item, and there is no `WM_MENUCHAR` message. Your information icon bitmap menu item is a second-class citizen.

Sorry.

Raymond Chen

Follow